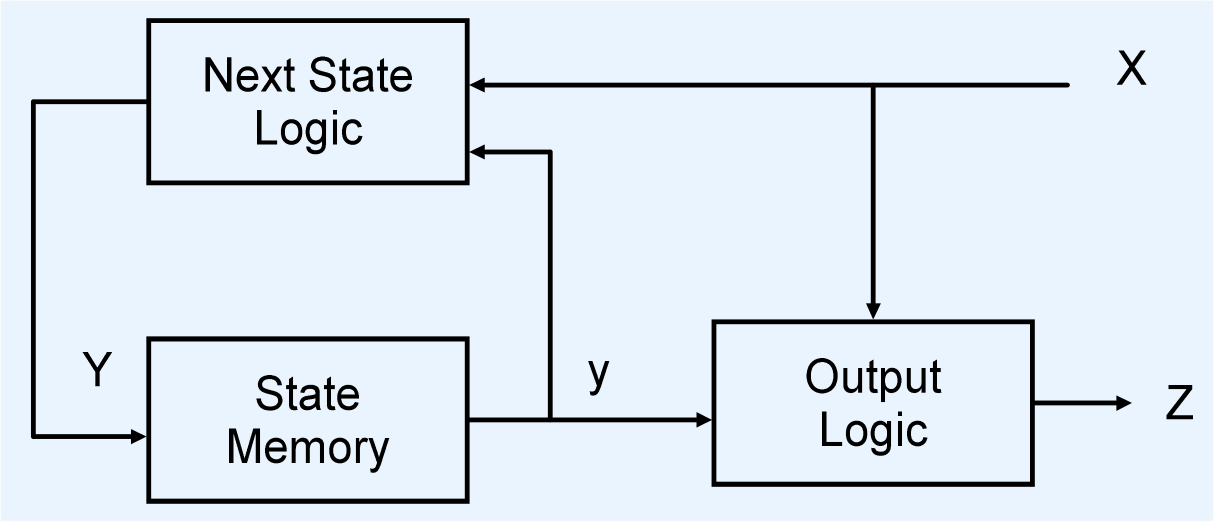
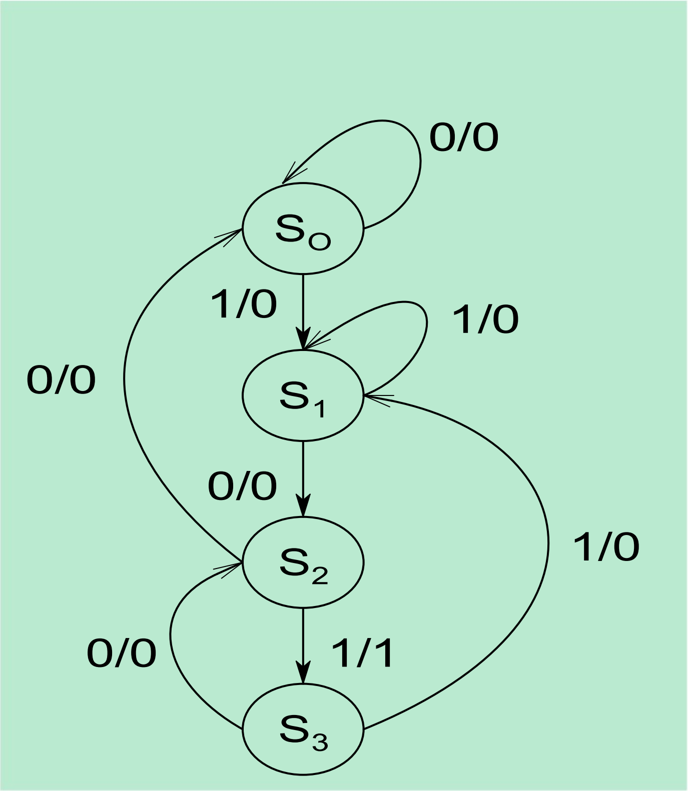
ESC201T : Introduction to Electronics

Lecture 39: Sequential circuit design-3

B. Mazhari
Dept. of EE, IIT Kanpur

Sequential Circuits: Summary

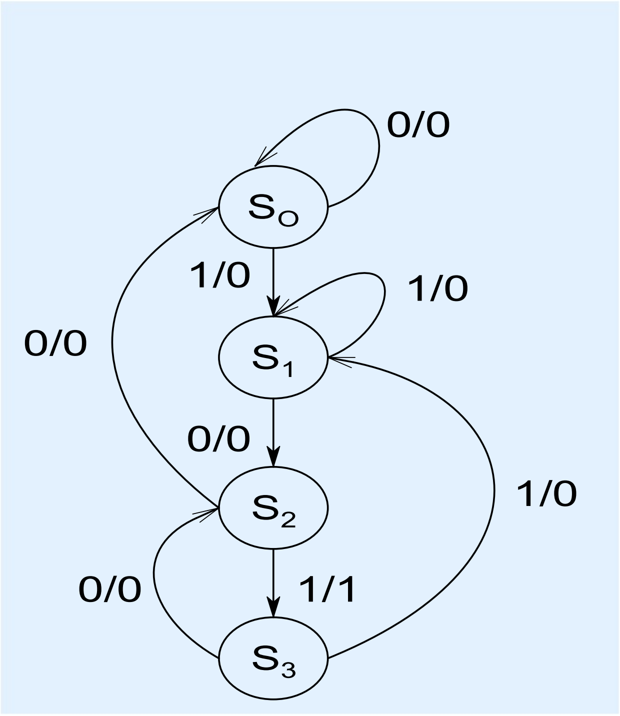


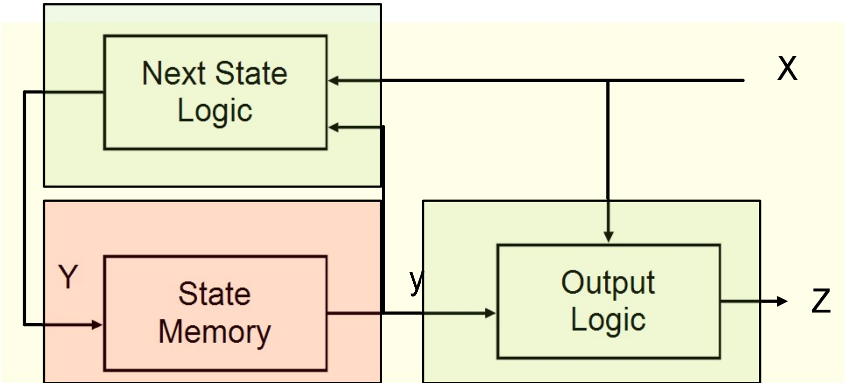


- we need to know how the system goes from one state to another in response to the inputs and how the outputs respond to these changes

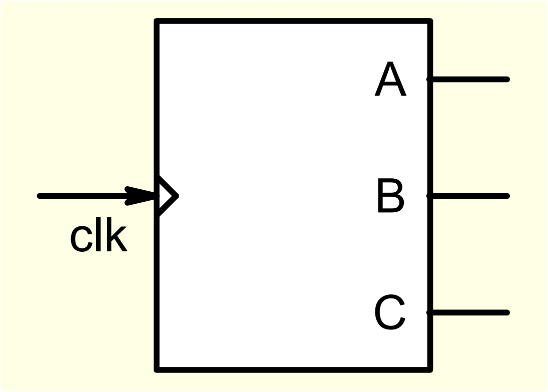
Synthesis involves the following tasks:

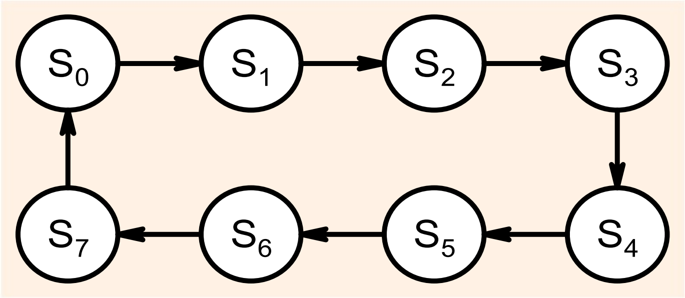
- Number of storage elements
- State Encoding
- Choosing a flipflop type to implement states
- Synthesize next state logic
- Synthesize output logic





Counters

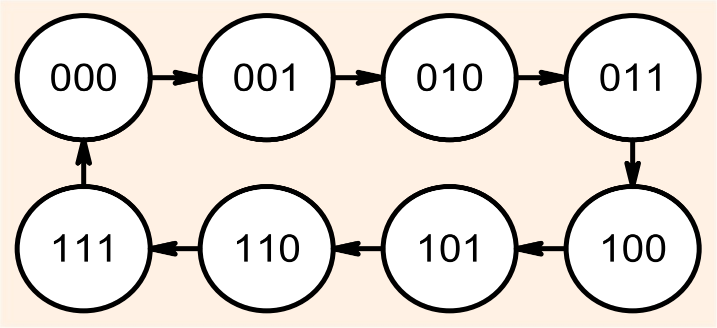




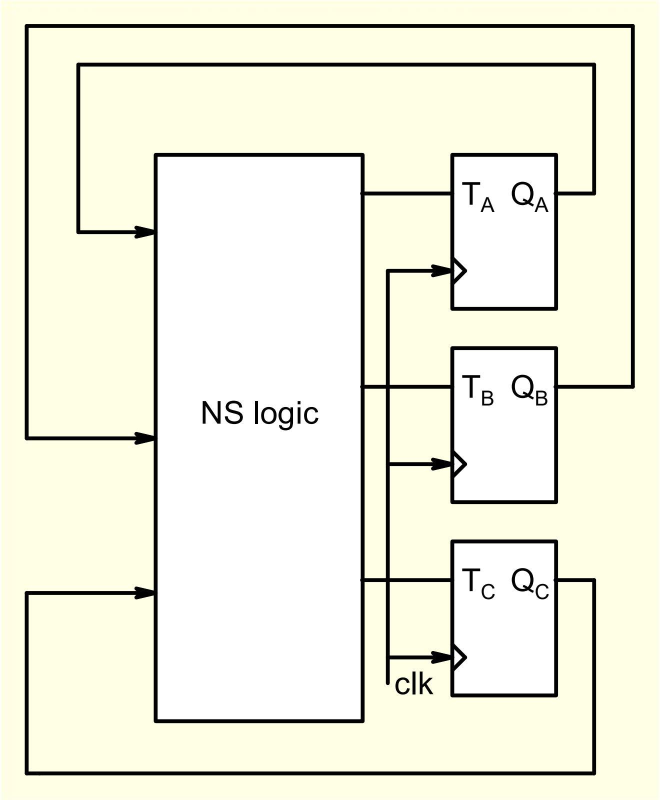
A	B	C
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

In state S_0, the output ABC is 000, in S_1 001 and so on

There are 8 states so 3 FFs are at least required. Let us choose T FF.

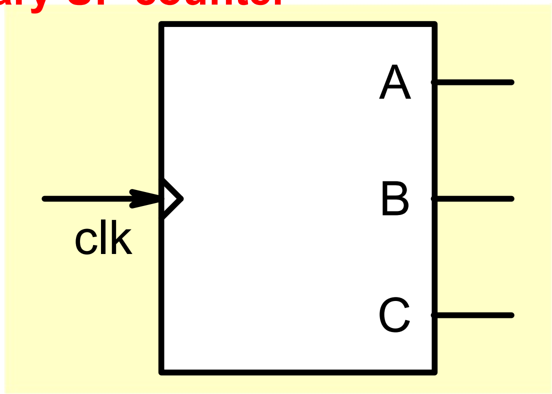


PS A B C	NS A B C	$T_A\ T_B\ T_C$
0 0 0	0 0 1	0 0 1
0 0 1	0 1 0	0 1 1
0 1 0	0 1 1	0 0 1
0 1 1	1 0 0	1 1 1
1 0 0	1 0 1	0 0 1
1 0 1	1 1 0	0 1 1
1 1 0	1 1 1	0 0 1
1 1 1	0 0 0	1 1 1

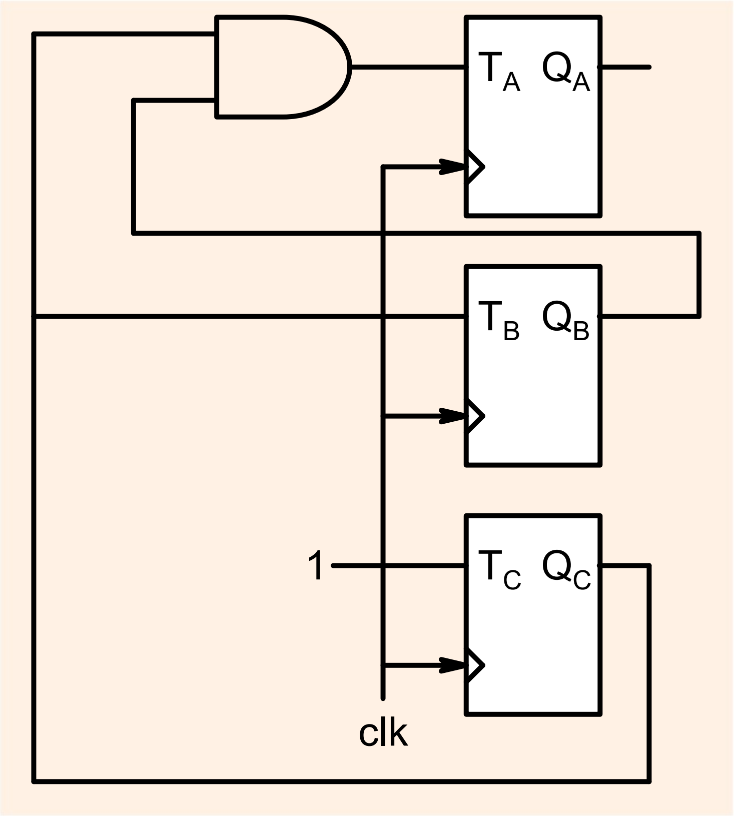


$$T_A = B.C\ ;\ T_B = C\ ;\ T_C = 1$$

Binary UP counter

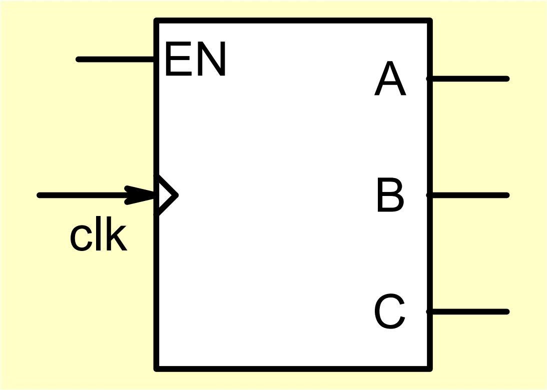


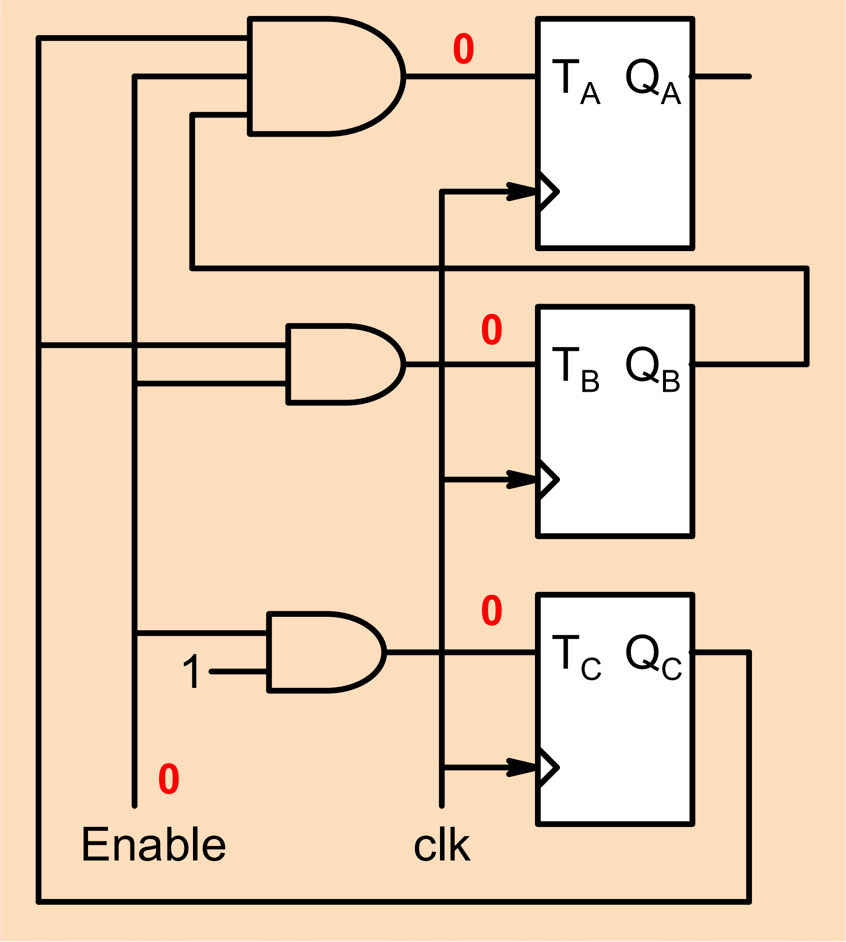
A	B	C
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1



$$T_A = B.C \;;\; T_B = C \;;\; T_C = 1$$

Counter with Enable

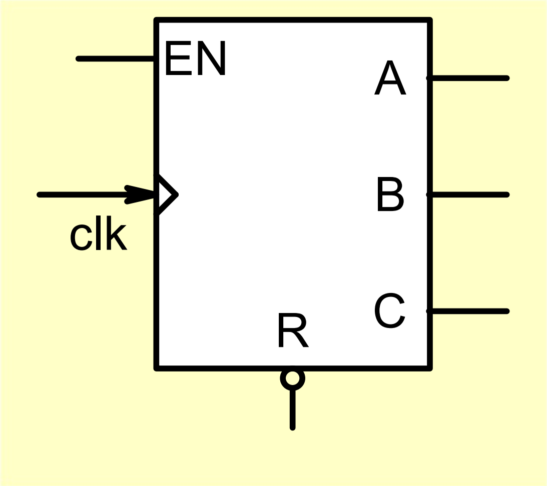


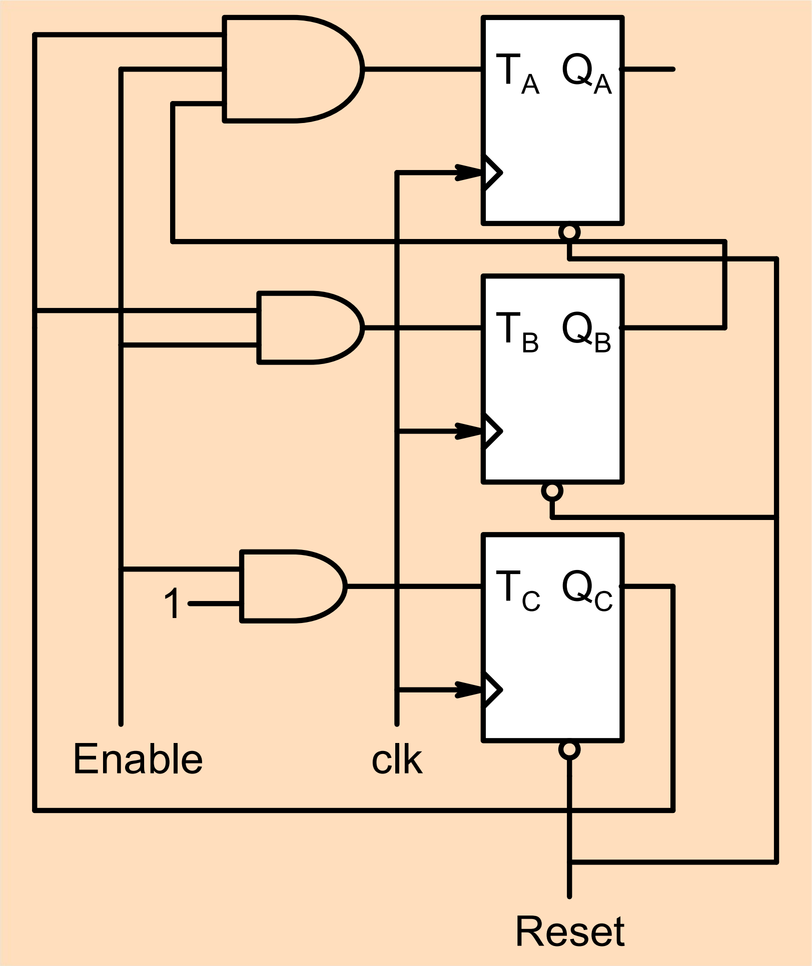


Counter is in Hold state.

When Enable = 1, the counter begins the count.

Counter with Asynchronous Reset



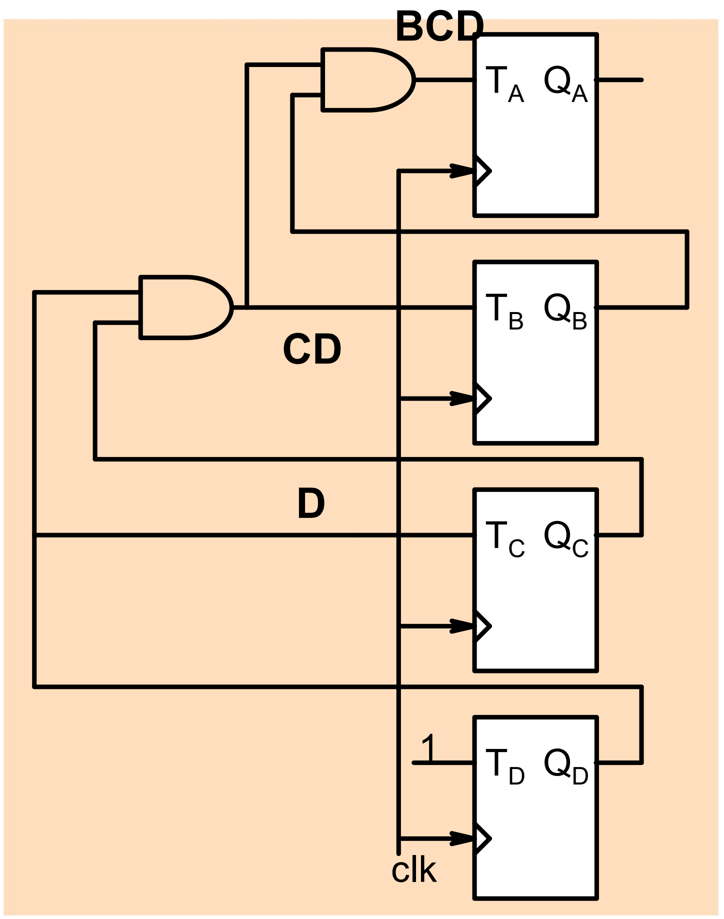


When Enable = 1, the counter begins the count.

A	B	C	D
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1
0	0	0	0

-D toggles every clock cycle
-B toggles only when D is 1
-C toggles only when both C and D are 1
-A toggles only when B C D are 1

T FF toggles when T=1



4-bit Down Counter

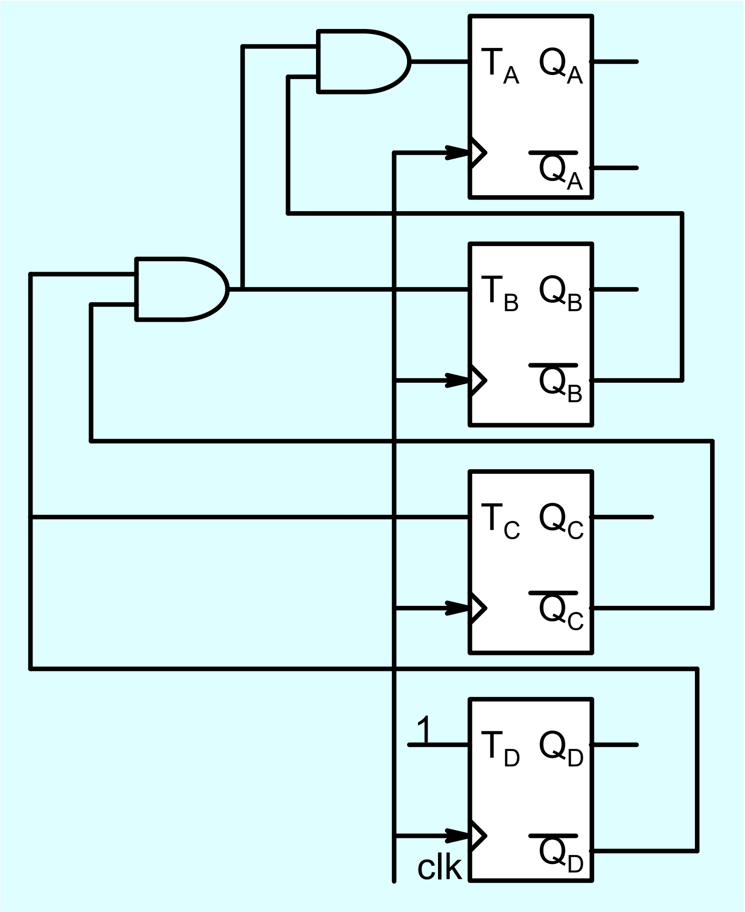
A	B	C	D
1	1	1	1
1	1	1	0
1	1	0	1
1	1	0	0
1	0	1	1
1	0	1	0
1	0	0	1
1	0	0	0
0	1	1	1
0	1	1	0
0	1	0	1
0	1	0	0
0	0	1	1
0	0	1	0
0	0	0	1
0	0	0	0
1	1	1	1

-D toggles every clock cycle

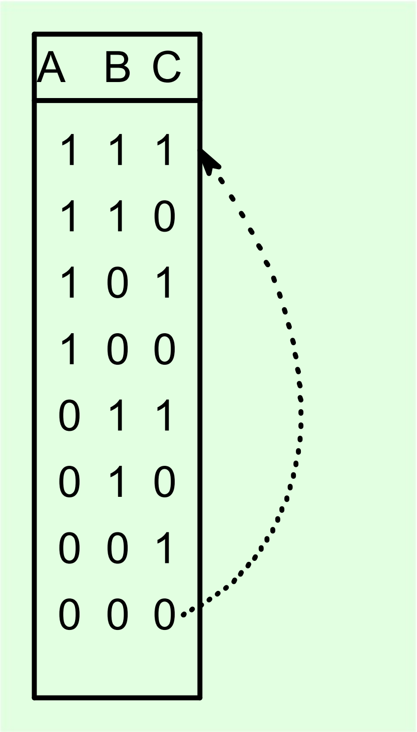
-C toggles only when D is 0

-B toggles only when both C and D are 0

-A toggles only when D C B are 0

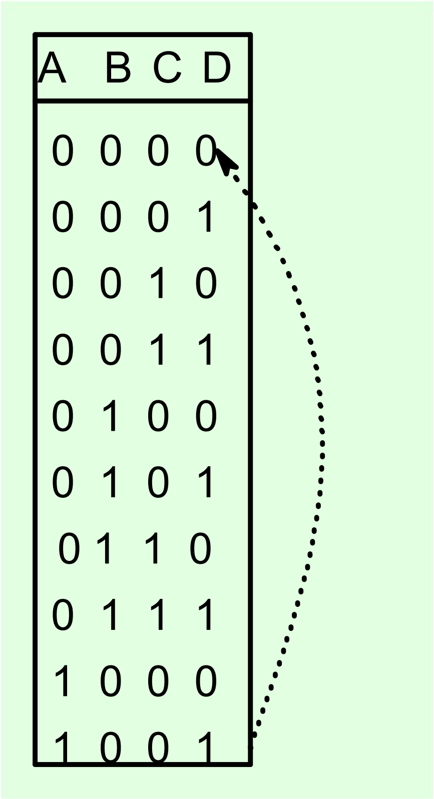


Counters



A	B	C
1	1	1
1	1	0
1	0	1
1	0	0
0	1	1
0	1	0
0	0	1
0	0	0

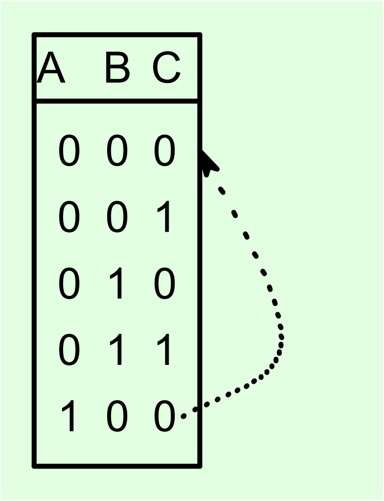
Binary down counter



A	B	C	D
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1

Decade counter

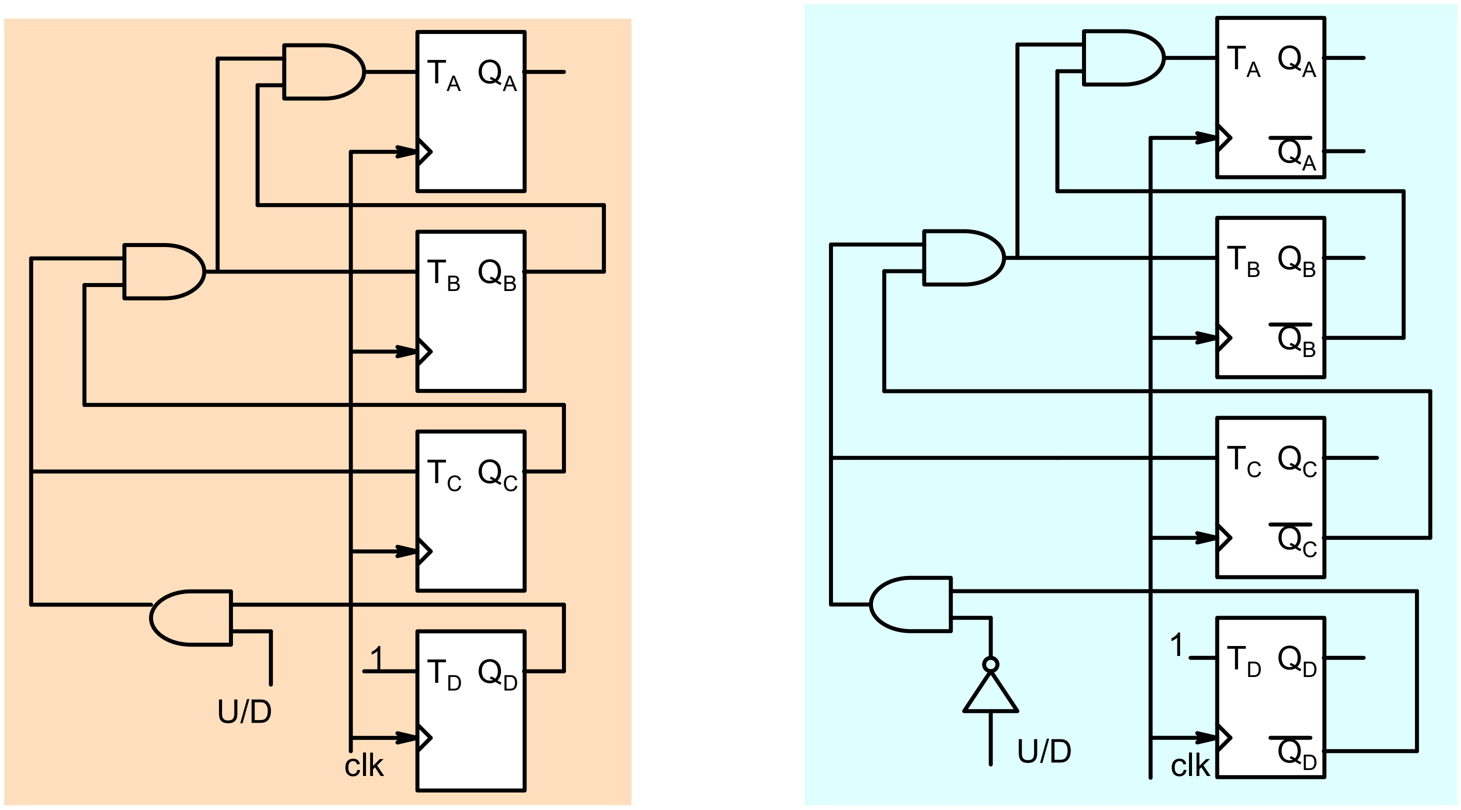
Modulo-10 Counter



A	B	C
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0

Modulo-5 Counter

4-bit Up-Down Counter

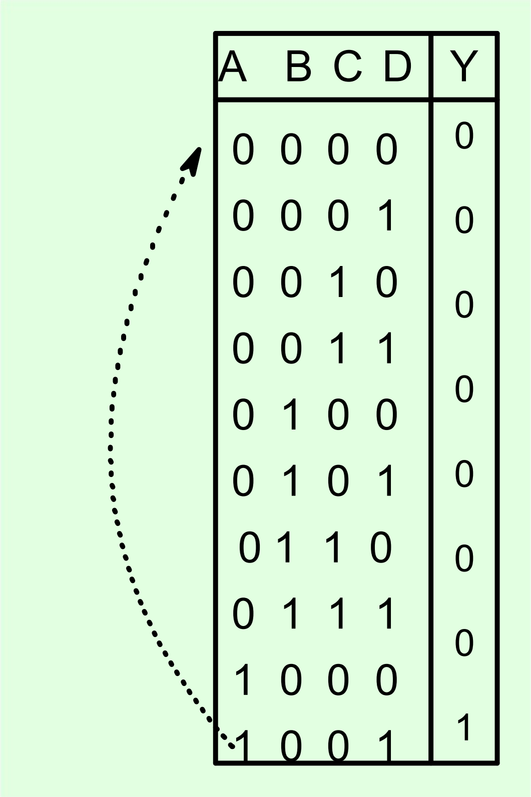

T_A Q_A
T_B Q_B
T_C Q_C
1
T_D Q_D
U/D
clk
T_A Q_A $\overline{Q}_A$
T_B Q_B $\overline{Q}_B$
T_C Q_C $\overline{Q}_C$
1
T_D Q_D $\overline{Q}_D$
U/D
clk

Merging of the two structures gives an Up/down counter

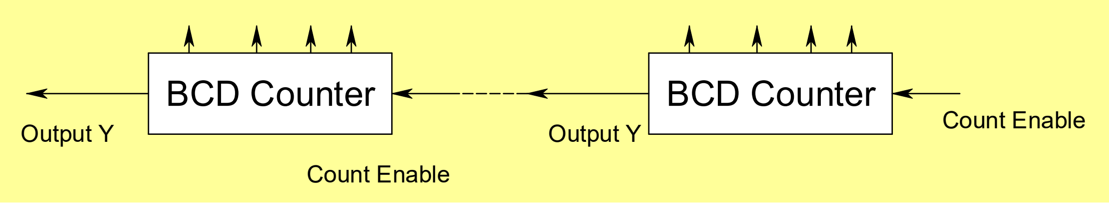
BCD Counter

Binary Coded Decimal (BCD): each decimal digit is coded as a 4-bit binary number

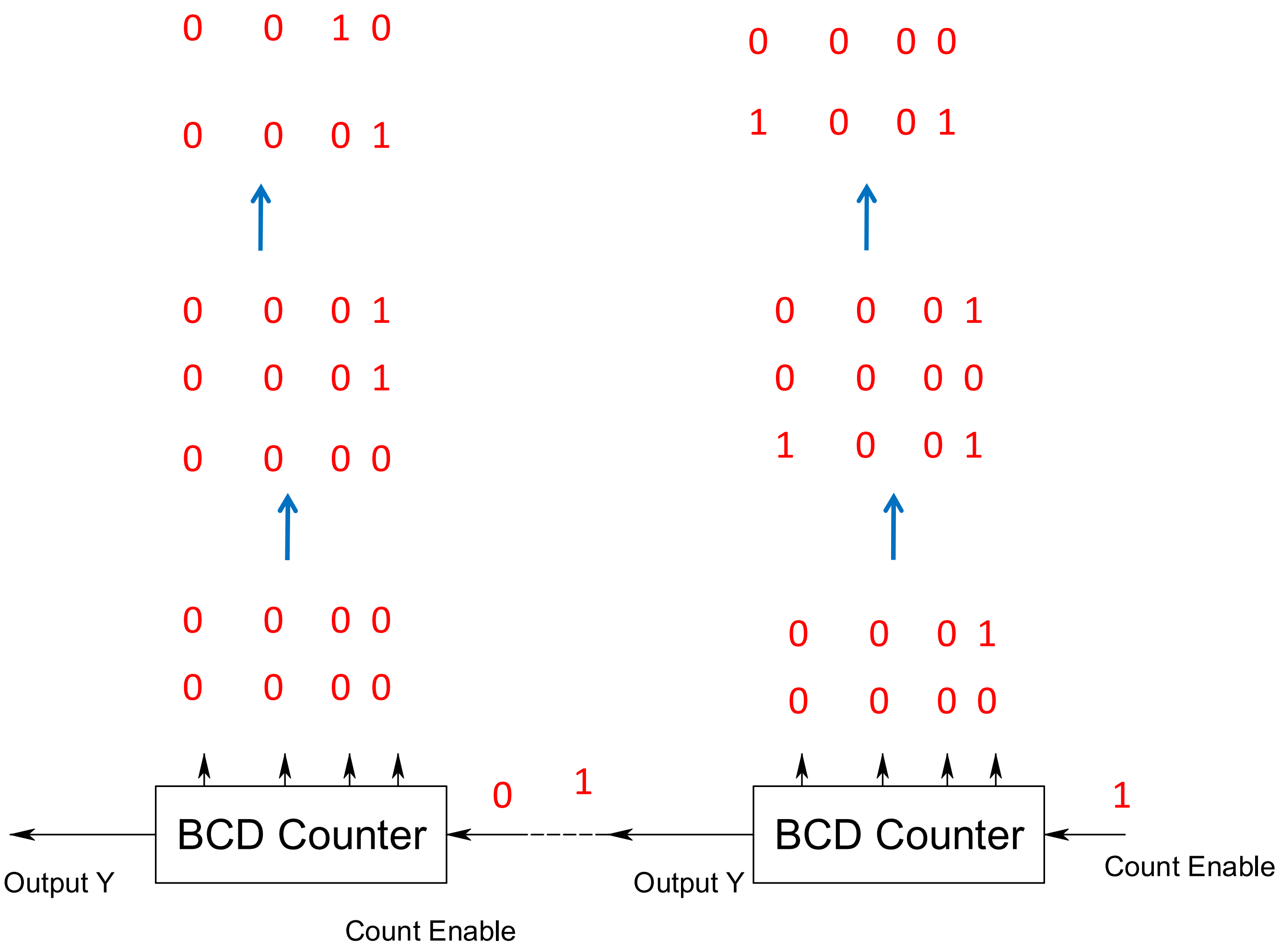
25 = 0010 0101



A	B	C	D	Y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1



BCD counter from 0 to 99



Counter with Unused States

PS			NS								
A	B	C	A	B	C	J_A	K_A	J_B	K_B	J_C	K_C
0	0	0	0	0	1	0	X	0	X	1	X
0	0	1	0	1	0	0	X	1	X	X	1
0	1	0	1	0	0	1	X	X	1	0	X
1	0	0	1	0	1	X	0	0	X	1	X
1	0	1	1	1	0	X	0	1	X	X	1
1	1	0	0	0	0	X	1	X	1	0	X

There are two unused states 011 and 111. one approach to handle this situation is that, while evaluating expressions for J K , we use don’t care conditions corresponding to these unused states

Counter with Unused States

PS			NS								
A	B	C	A	B	C	J_A	K_A	J_B	K_B	J_C	K_C
0	0	0	0	0	1	0	X	0	X	1	X
0	0	1	0	1	0	0	X	1	X	X	1
0	1	0	1	0	0	1	X	X	1	0	X
1	0	0	1	0	1	X	0	0	X	1	X
1	0	1	1	1	0	X	0	1	X	X	1
1	1	0	0	0	0	X	1	X	1	0	X

A \ BC	00	01	11	10
0	0	0	X	1
1	X	X	X	X

$J_A = B$

Counter with Unused States

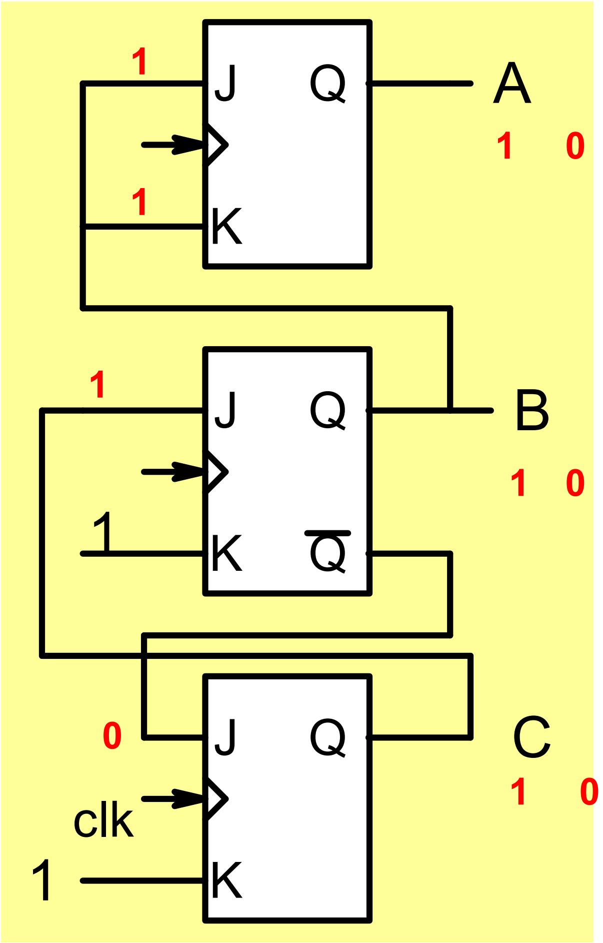
PS			NS								
A	B	C	A	B	C	J_A	K_A	J_B	K_B	J_C	K_C
0	0	0	0	0	1	0	X	0	X	1	X
0	0	1	0	1	0	0	X	1	X	X	1
0	1	0	1	0	0	1	X	X	1	0	X
1	0	0	1	0	1	X	0	0	X	1	X
1	0	1	1	1	0	X	0	1	X	X	1
1	1	0	0	0	0	X	1	X	1	0	X

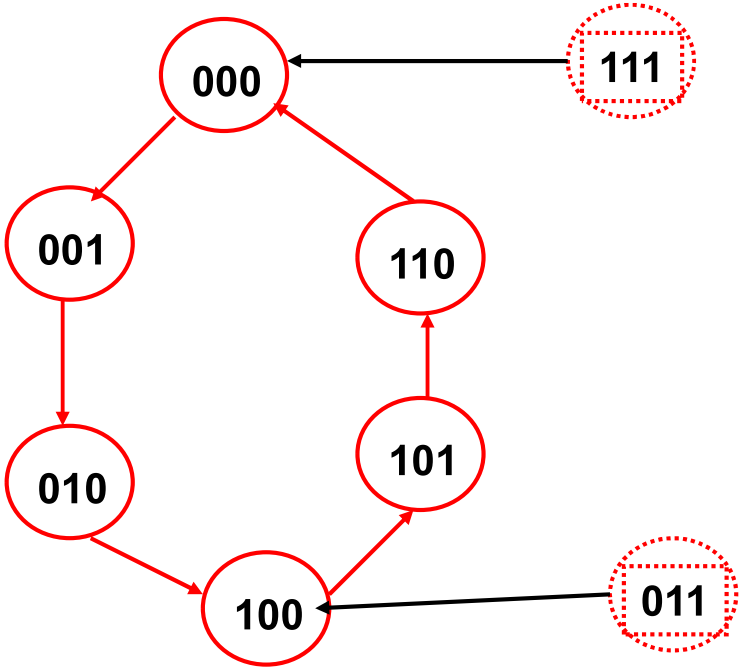
$$J_A = B \qquad K_A = B$$

$$J_B = C \qquad K_B = 1$$

$$J_C = \overline{B} \qquad K_C = 1$$

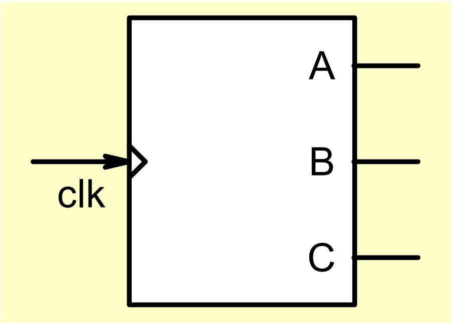
After synthesizing the circuit, one needs to check that if by chance the counter goes into one of the unused states, after one or more clock cycles, it enters a used state and then remains among the used states



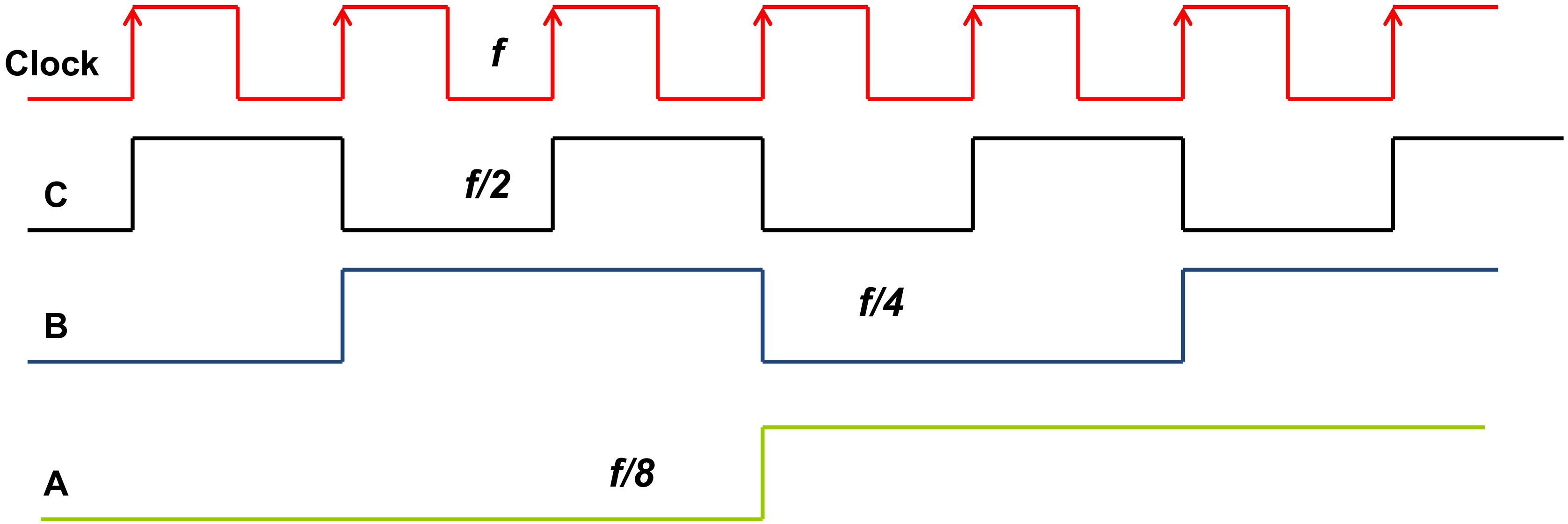


We can see that if by chance the counter goes into unused states 111 or 011, then after a clock cycle it enters one of the used states.

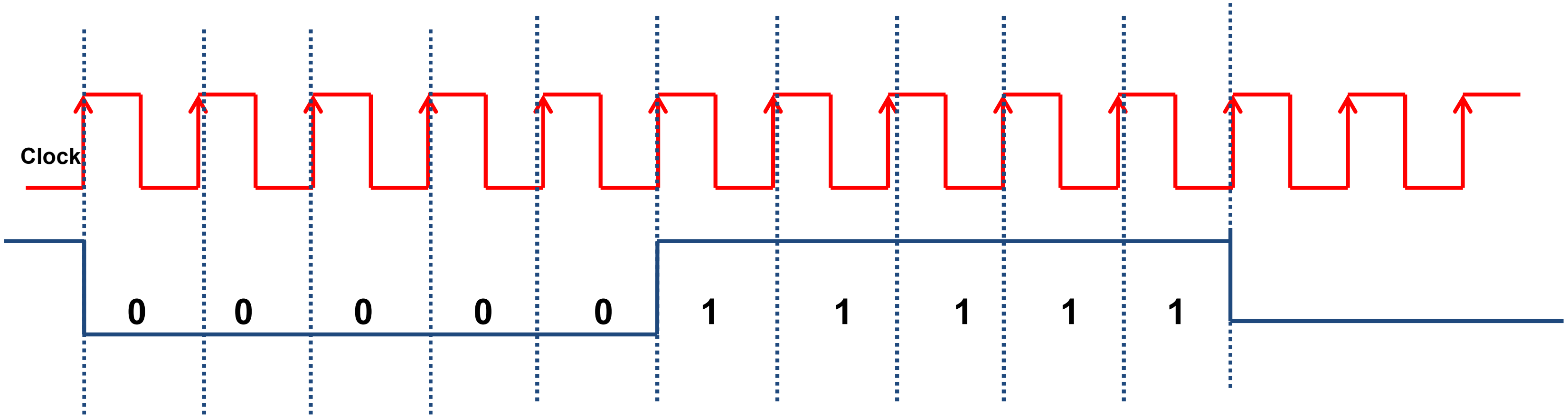
Counter as frequency divider



A	B	C
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1



Example From a frequency of 10KHz, generate the following signal of frequency 1KHz

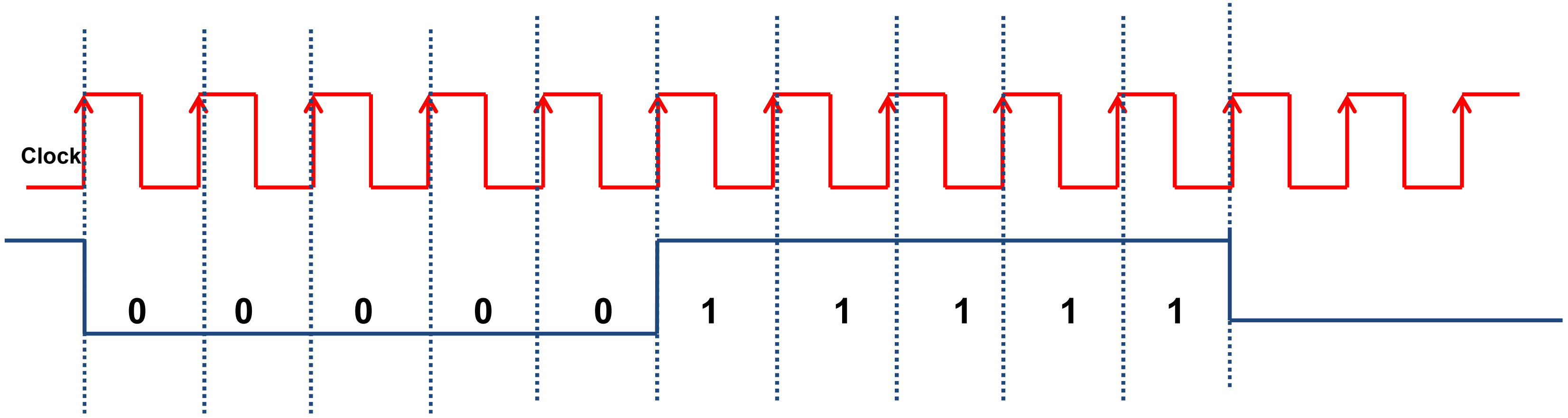


A	B	C	D
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1

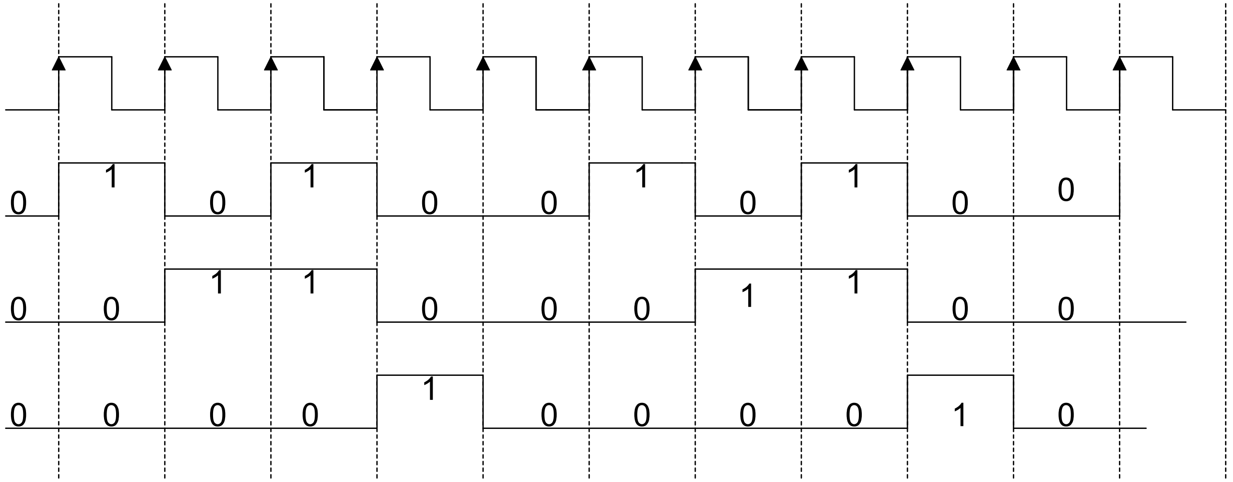
This will have a frequency of 1KHz but it will not have the same waveform

Example

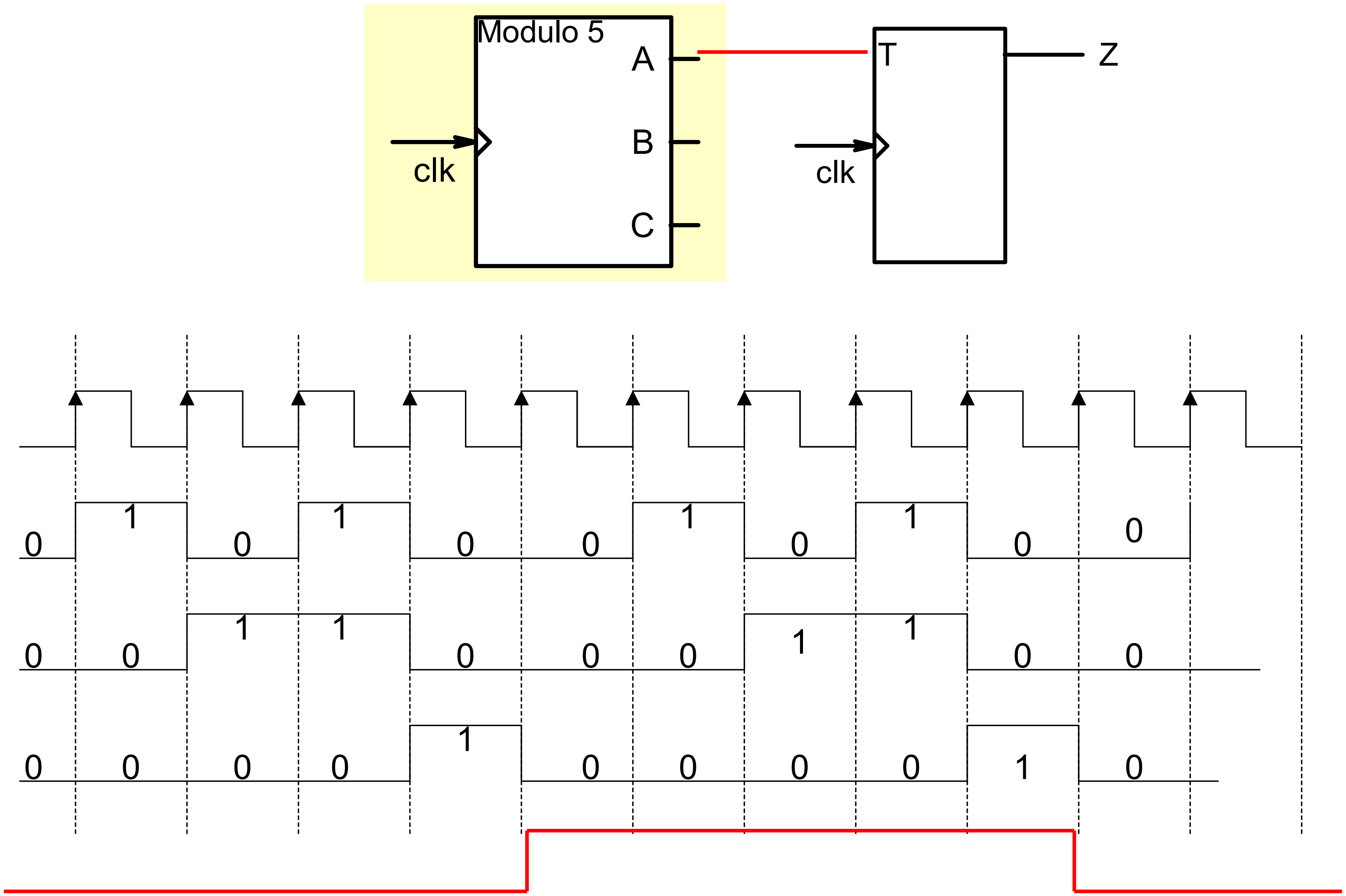
From a frequency of 10KHz, generate a signal of frequency 1KHz

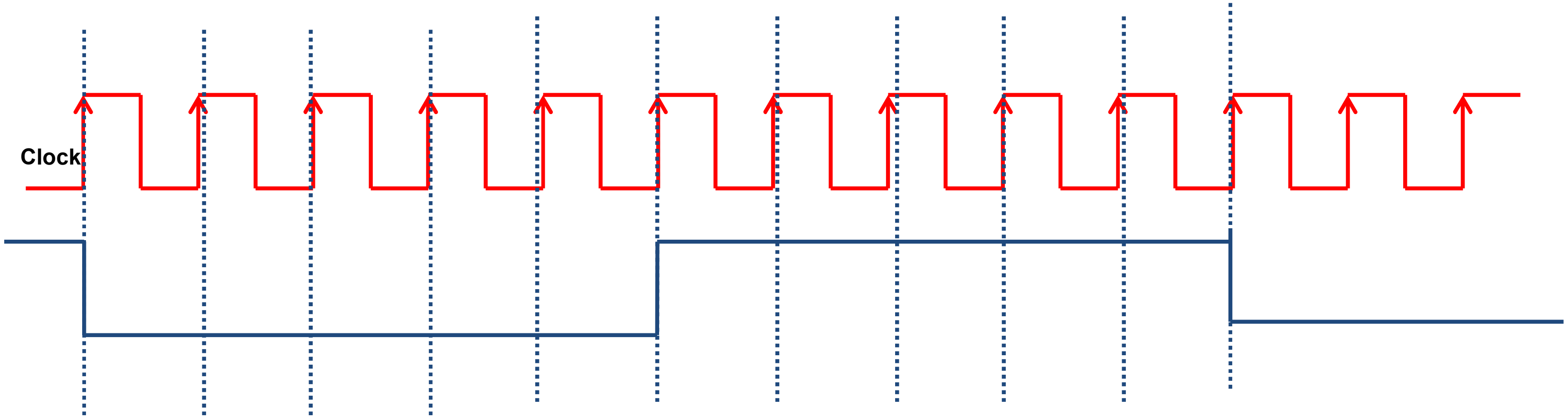


A	B	C
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0



The idea is to generate a divide by 5 counter first and then divide it again by 2 to get the required waveform

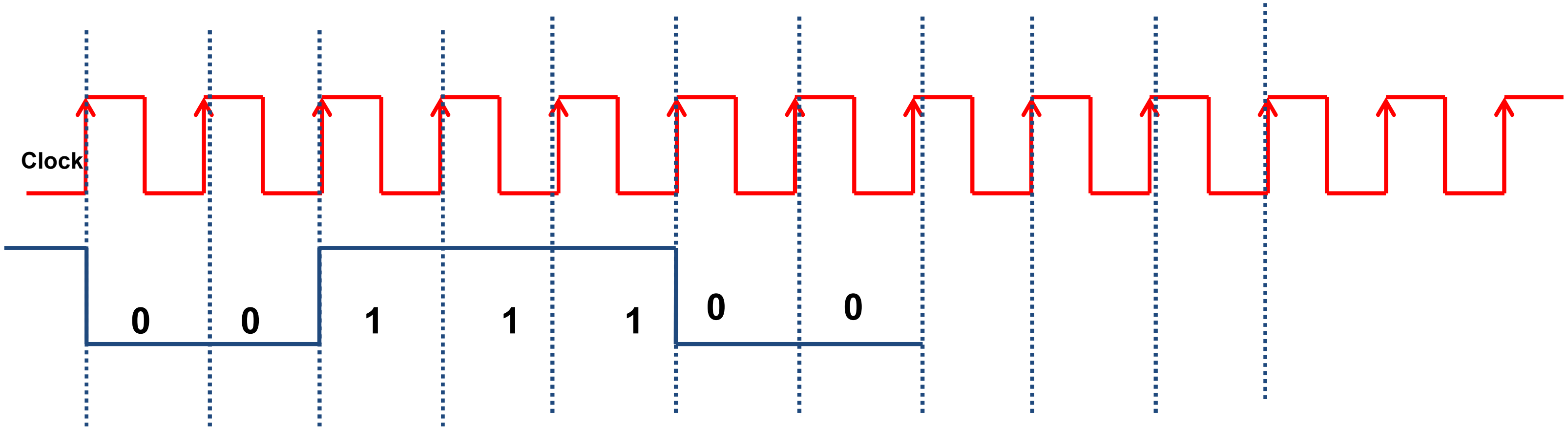

Modulo 5
A
B
C
clk
T
clk
Z
0 1 0 1 0 0 1 0 1 0 0
0 0 1 1 0 0 0 1 1 0 0
0 0 0 0 1 0 0 0 0 1 0



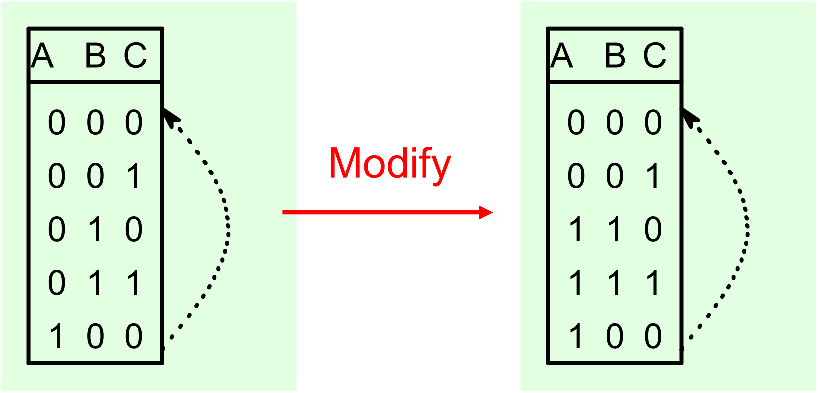
Alternatively design a divide by 10 counter with the following states

A	B	C	D
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0

Example From a frequency of 10KHz, generate the following signal of frequency 2KHz



A divide by 5 counter is required that has 5 states.



A	B	C
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0

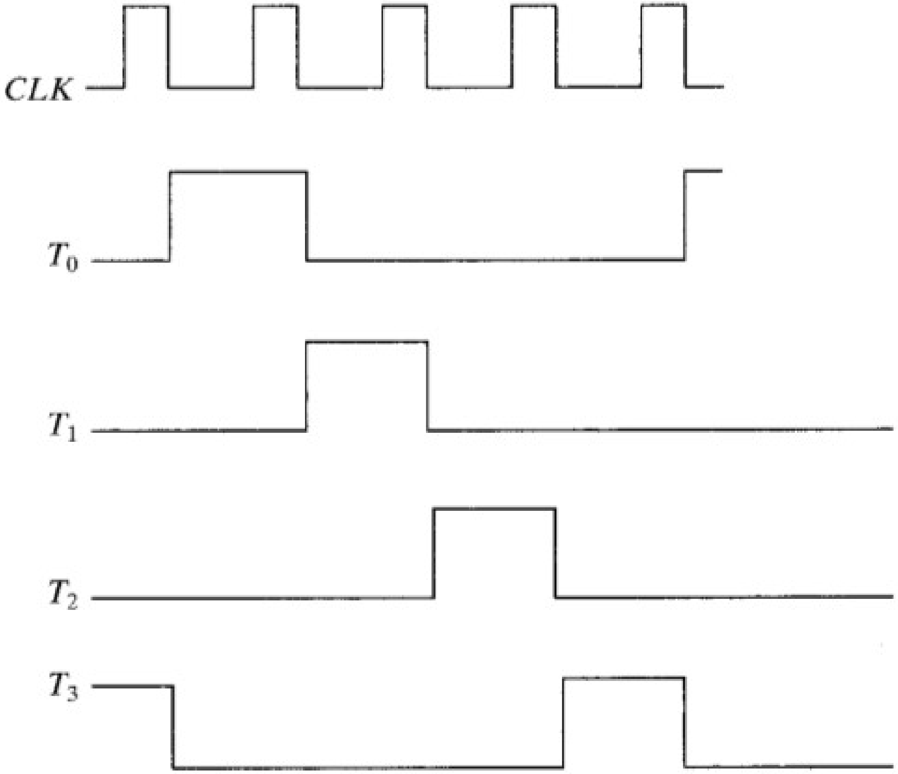
Modify

A	B	C
0	0	0
0	0	1
1	1	0
1	1	1
1	0	0

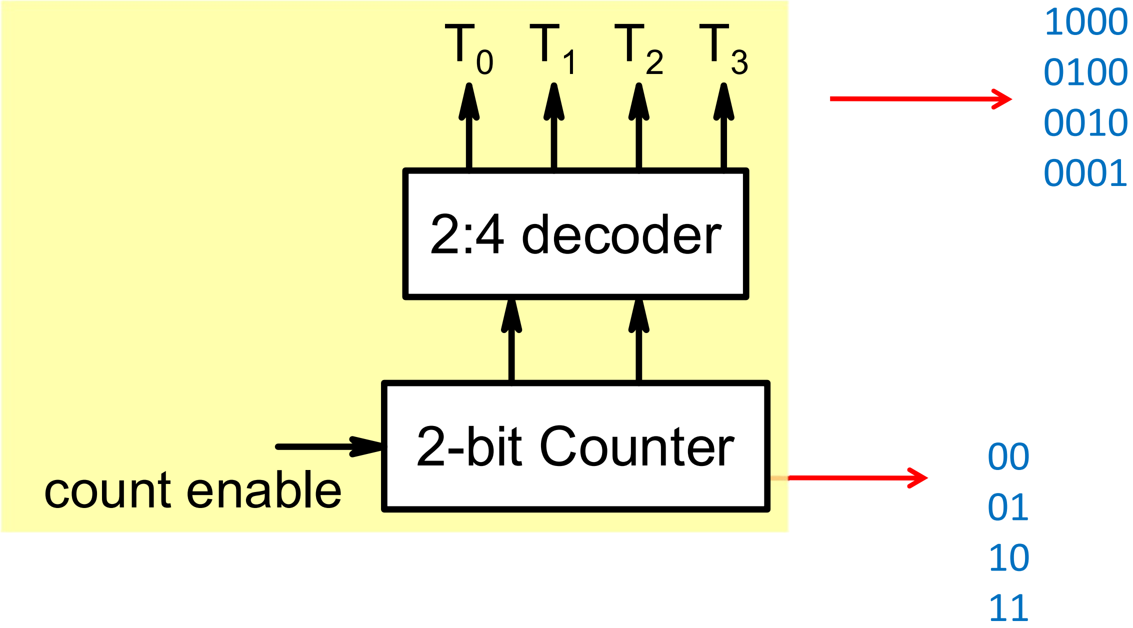
A will give the required waveform.

Ring Counter

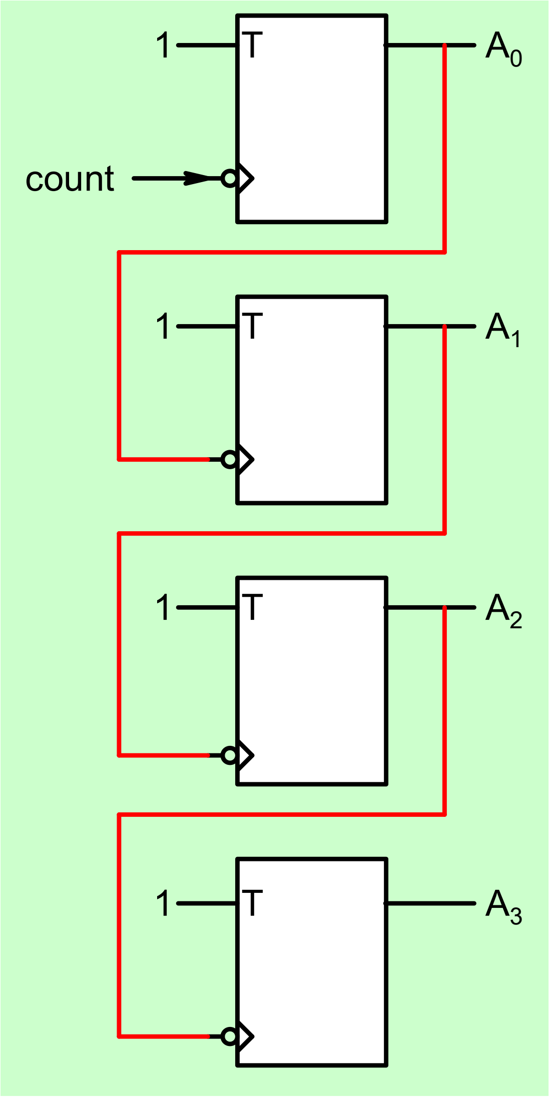
T_3	T_2	T_1	T_0
0	0	0	1
0	0	1	0
0	1	0	0
1	0	0	0



Alternative Implementation



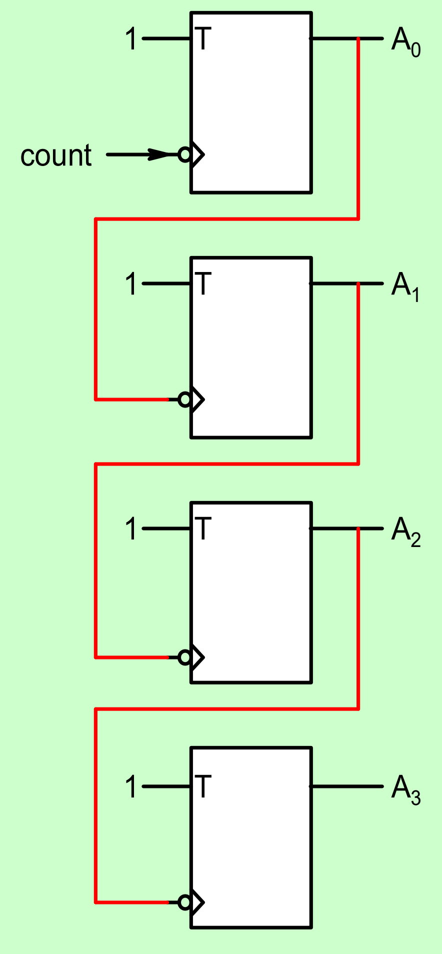
Ripple Counter



T FF toggles when T = 1; otherwise Hold state

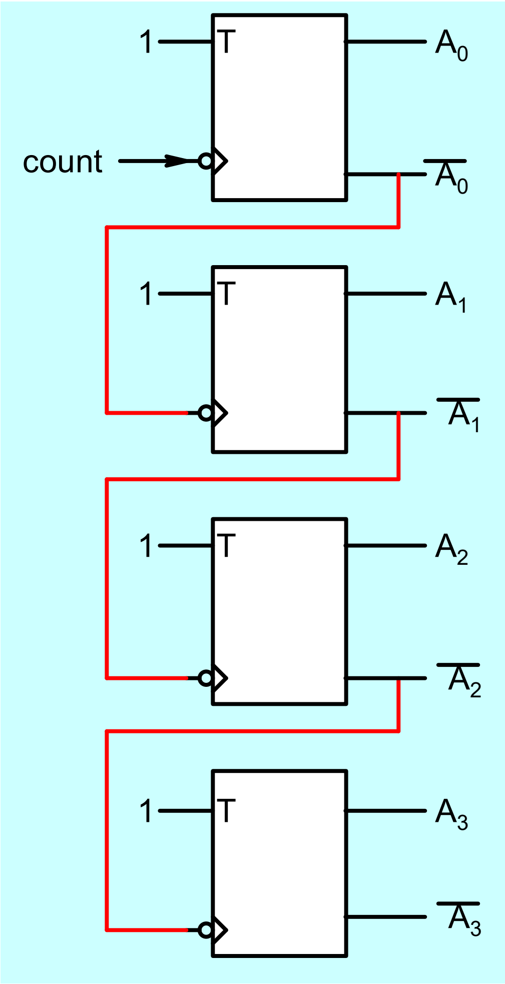
Clock is negative edge Triggered

Ripple Counter



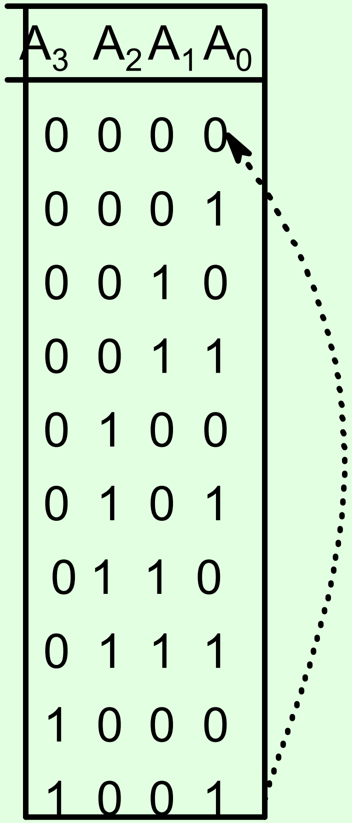
0	1	2	3	4	5	------15	
0	1	0	1	0	1	------1	0
0	0	1	1	0	0	------1	0
0	0	0	0	1	1	------1	0
0	0	0	0	0	0	------1	0

Ripple Down Counter

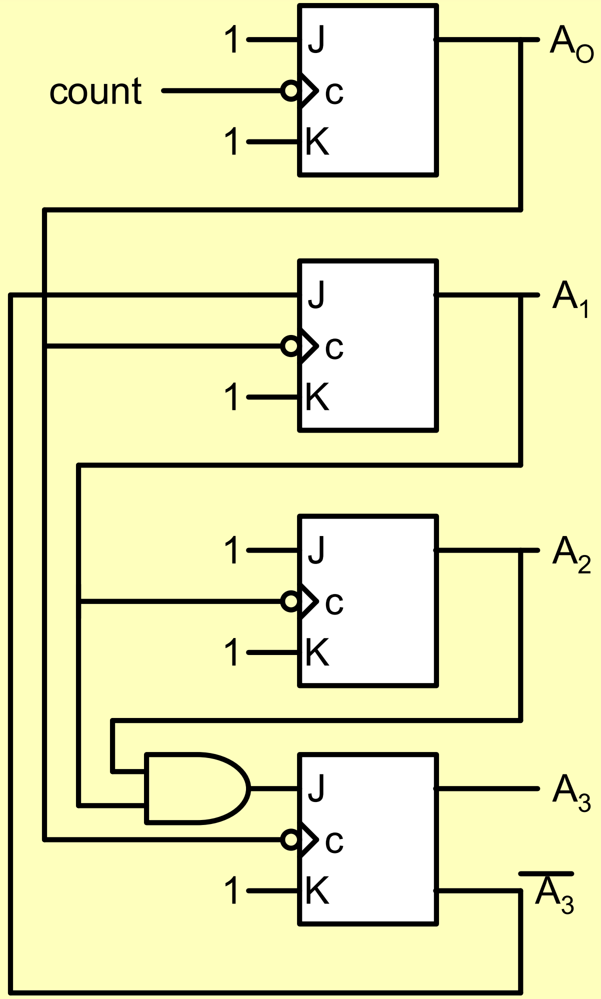


0	1	0
0	1	1
0	1	1
0	1	1

BCD Ripple Counter

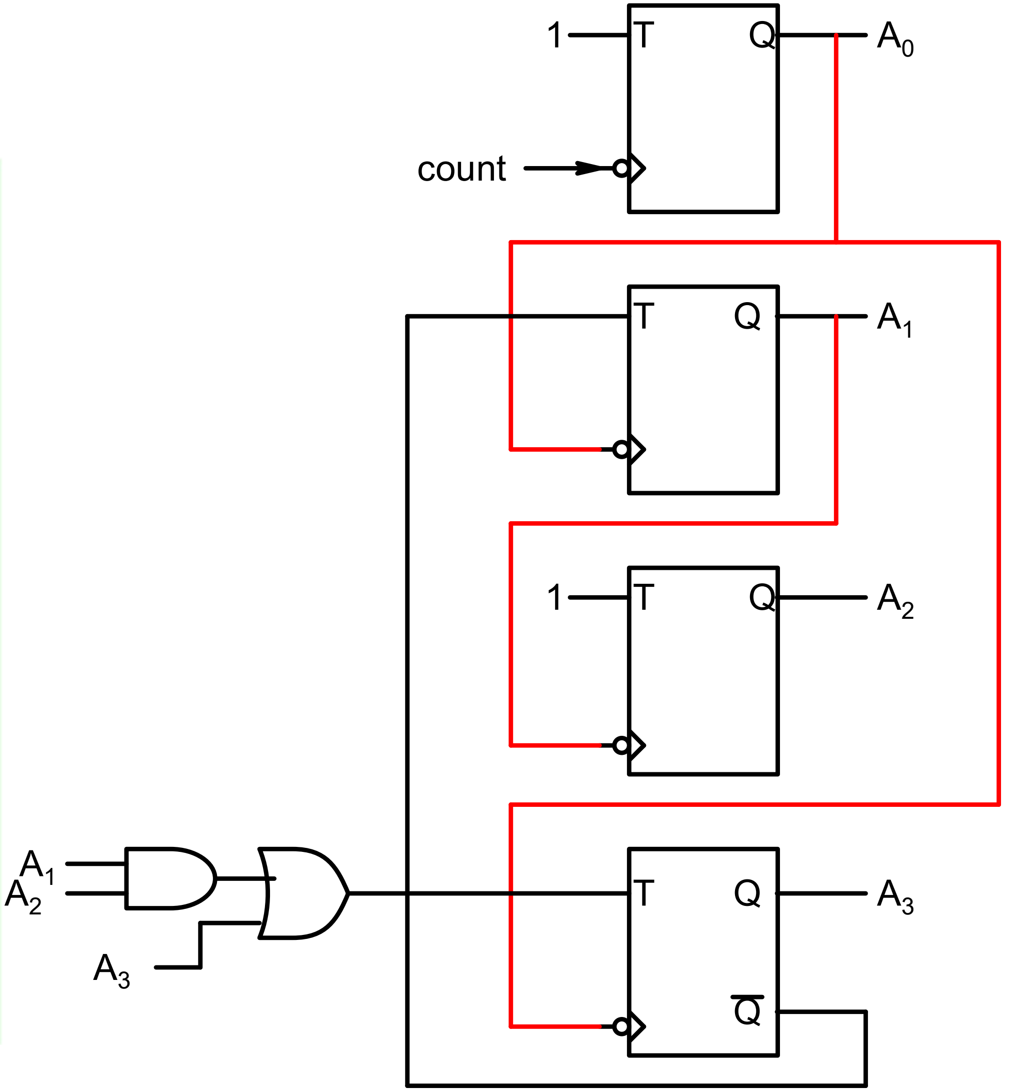


A_3	A_2	A_1	A_0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1

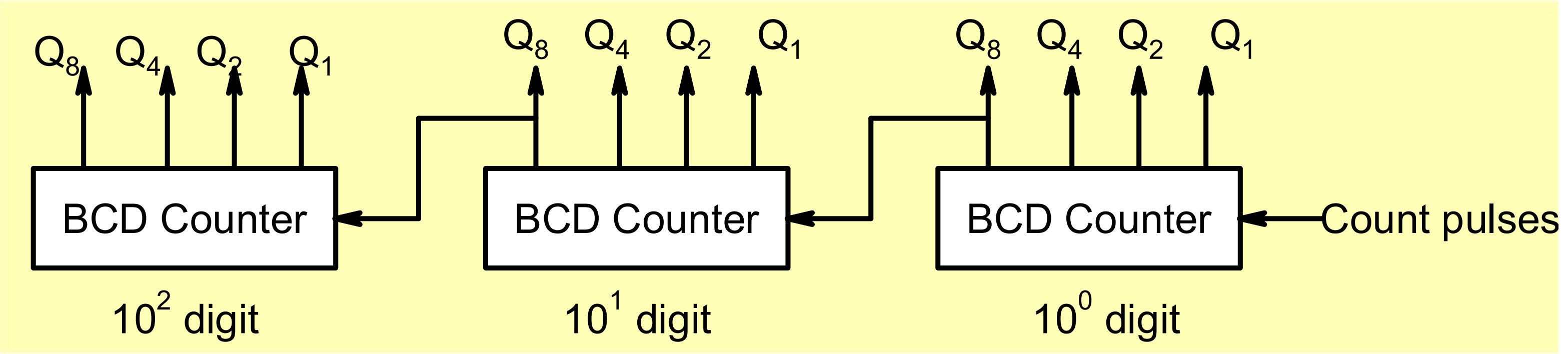


BCD Ripple Counter

A_3	A_2	A_1	A_0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1



Cascading of BCD counters


Q8 Q4 Q2 Q1
BCD Counter
10² digit
Q8 Q4 Q2 Q1
BCD Counter
10¹ digit
Q8 Q4 Q2 Q1
BCD Counter
10⁰ digit
Count pulses